20000920303

AD-A262 360

ITATION PAGE

Form Approved
OMB No. 0704-0188

...T DATE	3. REPORT TYPE AND DATES COVERED
	Final Report 01 Jul 89 15 Sep 92

4. TITLE AND SUBTITLE
Random-like interconnects, fault tolerance and grain-size studies for optoelectronic computing

5. FUNDING NUMBERS
2305/DS

6. AUTHOR(S)
Professors Esener, Paturi, Lee

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Univ of California, San Diego
La Jolla, CA 92093

8. PERFORMING ORGANIZATION REPORT NUMBER
AFOSR-TR- 93 0188

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NE
110 Duncan Avenue, Suite B115
Bolling AFB, DC 20332-0001

Dr Alan E. Craig

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFOSR-89-0440

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
UNLIMITED

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Our objective during the funding period, July 1, 1989 to September 15, 1992, was to investigate random like interconnects, fault tolerance, and grain size studies for optoelectronic parallel processors. The major focus has been in the design and analysis of paralle optoelectronic interconnection networks. Two major areas were identified and researched. The first involves the design, analysis, and simulation of perfect shuffle-based optoelectronic multistage interconnection interconnection networks(MINs) for highly parallel computers. The objective was first to perform a quantitative performance comparison between optoelectronic and VLSI implementations of multistage interconnection networks (MINs). The next task was to optimize the optoelectronic MIN with respect to archtiectural and technological parameters. The final goal was to design and simulate a MIN system that could provide a complete set of communiction and synchronization services.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASS	UNCLASS	UNCLASS	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Final Progress Report

for

Random-like Interconnects, Fault Tolerance and Grain-Size Studies for Optoelectronic Computing

Sponsored by

Air Force Office of Scientific Research

Under Grant No AFOSR - 30602 - 89 - 0440

for Period 7/1/89 through 9/15/92

Grantee

The Regents Of the University of California, San Diego

University of California , San Diego

La Jolla CA 92093

Principal Investigators : S. C. ESENER (619) 534-2732,

R. PATURI (619) 534-6658

S.H. LEE (619) 534-2413,

Program Manager : Dr. A. CRAIG (202) 767-4931

Table of Contents

Random-like interconnects, fault tolerance and grain size studies for optoelectronic computing

Participating Personnel: Students: A. V. Krishnamoorthy, D. T. Li, P. J. Marchand, F. E. Kiamilev, G. C. Marsden
Researchers: P. J. Marchand

1. OBJECTIVES

Our objective during the funding period, July 1 1989 to September 15 1992, was to investigate random like interconnects, fault tolerance, and grain size studies for optoelectronic parallel processors. The major focus has been in the design and analysis of parallel optoelectronic interconnection networks. Two major areas were identified and researched. The first involves the design, analysis, and simulation of perfect shuffle-based optoelectronic multistage interconnection interconnection networks (MINs) for highly parallel computers. The objective was first to perform a quantitative performance comparison between optoelectronic and VLSI implementations of multistage interconnection networks (MINs). The next task was to optimize the optoelectronic MIN with respect to architectural and technological parameters. The final goal was to design and simulate a MIN system that could provide a complete set of communication and synchronization services.

The second area of concentration involved the design and simulation of fault tolerant optoelectronic interconnection networks based on random-like interconnection networks. The first task was to quantify the inherent tolerance to hardware faults provided by networks based on random interconnects. Next, engineering tradeoffs involving the balance between the grain size, the optical interconnect complexity, and the running time were studied. The final objective was to demonstrate the feasibility of parallel testing and run-time fault tolerance on a fully functional computer model of Programmable Opto-Electronic Multiprocessor (POEM) system. We are currently combining these result into an optoelectronic design and implementation of a fault-tolerant low-contention interconnection network called 2-Butterfly or Twin Butterfly.

2. GRAIN SIZE STUDIES

During this project we designed an optoelectronic interconnection network based on the well known perfect shuffle network topology. The shuffle-exchange network (Omega network) is one of many isomorphic networks known as Banyan networks. Figure 1 shows an example of a self-routing Banyan network with 16 input/output channels. Our work included the detailed design of the optoelectronic system, the layout of the optoelectronic chip, and the gate level design and simulation of the optoelectronic processing elements. To justify the implementation of a shuffle-based routing MIN, we first compared the cost and performance of optoelectronic and VLSI MIN implementations

2.1 Comparison to VLSI

To compare implementation technologies quantitatively, we chose the well-known perfect shuffle where both purely electrical and optoelectronic implementations exist. Implementations of equivalent multistage interconnection networks were compared in terms of footprint area, speed, network bandwidth, and power consumption. The results of the comparison show that for large

networks (N>256), optoelectronic outperforms VLSI in speed and bandwidth. Beyond N=256, the VLSI network bandwidth saturates, while the optoelectronic network bandwidth continues to increase. Furthermore, for networks with N>2048, the optoelectronic network has a smaller footprint area than the VLSI network. The footprint area for the optoelectronic network grows linearly with the number of switching elements, while the VLSI system area increases as the square of the number of switching elements. For large networks, the power consumption and the on-chip power density are higher for the optoelectronic network, because the VLSI network power drops off when the speed of operation is reduced to accommodate global wire delay.

The potential scalability of both technologies has also been analyzed. For the VLSI implementation, if the maximum chip size is 2cm x 2cm, then the maximum network size is N=128. Other limitations to VLSI network size include limited I/O pin count (N=512) and bandwidth saturation (N=256). For the optoelectronic implementation, the scalability is limited by the diffractive optical element width, the optoelectronic chip width, and the optical power requirements. The results show that the optoelectronic network is much more scalable than the VLSI network. The analysis was also extended to study the effect of variation of technology component parameters on the performance and comparison of both networks. The results show that the break-even point (i.e., the network size beyond which optoelectronics outperform VLSI) for bandwidth, speed, and footprint area are inversely proportional to the VLSI feature size. Figure 2 summarizes the effect of network size on the above mentioned parameters for both optoelectronic and VLSI multistage interconnection networks. Publication 1 provides the details.

2.2 Grain size optimization

We further modified our optoelectronic system design to allow the ratio of optics and electronic to be varied without affecting the functionality of the system. To accomplish this task we developed a novel optical system design (figure 3) and optoelectronic chip layout (figure 4) that allow the ratio of the electronic gates to the optical transmitters in the system to be optimized. The resulting design was then optimized with respect to the system cost and performance functions including the system volume, area, power consumption and bandwidth (figure 5). The result showed that an optimal optoelectronic MIN uses a switch size of K=64, corresponding to approximately 300 electronic gates per optical transmitter (figure 6). A detailed description of our design and our grain size optimization results are described in publication 2.

2.3 Effects of device parameters

To determine the effect of technology on the grain size results, we have also carried out detailed technology parameter variation studies (figure 7). The results show that improving electronic technology, for instance by reducing the minimum VLSI feature size, tends to increase the optimal grain size toward systems with a higher number of gates per optical I/O. Likewise, improving optoelectronic device performance tends to move the optimum to smaller grain size systems with more optical stages and fewer gates per optical I/O. The use of detailed modeling allowed us to determine the exact nature of these effects and provide feedback to the designers of optoelectronic devices and optical interconnects as to the key device improvements required for optimum performance. These results have been detailed in publications 2 and 5.

3. SMART NETWORK ARCHITECTURE

During the final funding period (May 1992 to September 1992) we focused our efforts toward the development of a "smart" interconnection network suitable for distributed computing. The initial part of our effort was to review the status of interconnection networks for distributed computing. We found that the performance bottleneck (e.g. synchronization bottleneck) of conventional network architectures in distributed computing environments has been previously recognized by the research community and several

architectural schemes have been proposed to address the problem. However, the proposed schemes do not scale well to interconnection networks with large numbers of processors (e.g. over 10,000 nodes). This occurs for several reasons:

1. Incompatibility with conventional network architectures and standards.
2. Higher design complexity and implementation cost than conventional network architectures.
3. Lack of efficient and scalable implementations with VLSI technology.

Based on these findings, our objective became to develop an interconnection network architecture that extends established networks to distributed computing and to an efficient implementation with optical interconnects.

3.1 Significance of smart network

The architectural part of our work involved extending the architecture of the familiar batcher-banyan network to provide synchronization services. The resulting network, called the **smart network**, provides the following advantages for distributed computing over previous designs:

1. The smart network design is downward compatible with a well-known network architecture (e.g. batcher-banyan). Thus, synchronization services can be introduced transparently, without disturbing the normal communication services of the network.
2. The cost of adding synchronization services to the batcher banyan network with multicast capability is small. The process of adding synchronization services is modular. This makes the incorporation of synchronization services attractive in cases where a batcher-banyan network is already being considered.
3. The serial bottleneck for synchronization operations associated with conventional networks is removed. With the proposed scheme, synchronization operations do not create output port blocking. Also performance is not dependent of the number of processors that participate in distributed computation.
4. The interconnection topology and the switching element design are fully compatible with our previous optically interconnected multistage interconnection network designs. Thus efficient and scalable implementation with optoelectronic technology is immediately possible.

The total cost of adding synchronization services to the batcher banyan network with multicast capability is very small. For example, adding a single synchronization service to a 1024 channel batcher banyan network would require 10 extra network stages ($\log_2 1024$), in addition to the 131 stages ($\log_2^2 1024 + 3\log_2 1024 + 1$) that are already in the network. While the cost of adding synchronization services to the batcher banyan is small, their effect on performance of distributed applications can be dramatic.

Typically, synchronization bottleneck arises when many messages (e.g. packets) are sent to the same destination port. In conventional networks, these packets serially enter the receiving processor and have some operation be performed on them. Usually the operation performed is commutative such as ORing, ADDing or ANDing the payloads of the packets. In the smart network, the network hardware performs the commutative operation within the network hardware. A single packet containing the result of the computation is delivered to the receiving processor.

The smart network design removes the serial bottleneck associated with previous designs. Moreover, the performance of synchronization operations is not dependent on the number of processors that request the service. The remainder of this section describes the smart

network architecture. To put our work in proper perspective, we will first review previous batcher-banyan network designs.

3.2 Batcher-banyan network with multicast capability

The batcher-banyan network is a non-blocking synchronous packet switch based on the shuffle interconnection topology. Previously, it has been proposed to serve as an asynchronous transfer mode (ATM) network for the emerging integrated digital network standard (ISDN).

The basic operation of the batcher-banyan network is to first sort the incoming packets according to their destination address using the batcher sorting network. For N bit-serial channels, the batcher network requires $\log_2^2 N$ stages of 2x2 switching elements to accomplish this task. This is followed by a single stage trap network that identifies and arbitrates packets headed to the same output destinations (i.e. output port contention). The trap network is followed by a banyan routing network that delivers the packets to their destination. This network requires $2\log_2 N$ stages switching elements.

The multicast service (also called broadcast or one-to-many communication) can be implemented by modifying the batcher banyan network. The basic idea is to use an additional batcher sorting network, called **group network**, to sort the incoming packets according to their group address. The group address is an additional field added to the packet header to implement user-initiated multicast operation. With this scheme, network channels that participate in a multicast operation (i.e. one master and many copy channels) send packets into the network with the same group address.

The group sorting network is followed by a network, called **copy network**, that copies the contents of the master packet to the copy packets. The copy network uses the banyan topology and requires $\log_2 N$ stages of 2x2 switching elements. The basic batcher banyan network is then used to route the packets to their destinations. To implement multicast service properly the copy packets must have their own port as the destination address. The total number of stages of 2x2 switching elements required to implement a batcher banyan network with multicast service is $\log_2^2 N + 3\log_2 N + 1$.

As previously stated, we have developed a simple architectural modification of the multicast-capable batcher banyan network to provide synchronization services. The basic idea behind our scheme is to insert additional networks between the group sorting network and the copy network. Each of these new networks implements a specific synchronization service and uses the banyan topology with $\log_2 N$ stages of 2x2 switching elements. For example, the network that performs the distributed ADD operation is shown in figure 8.

As in multicast operations, the group address is used to identify packets that participate in a particular synchronization service. Additional control bits are added to the packet header for each possible synchronization operation, with the idea that the networks performing synchronization operations check these status bits and the group address to determine whether the packet should participate in the synchronization operation. Figure 9 illustrates the packet format and the structure of the smart network.

3.3 VHDL simulation and synthesis

The final portion of this work has been to design and simulate a small-scale smart network with VHDL synthesis and simulation tools. Our simulation included synchronization

services for ADD, AND, OR and fetch-and-add. The entire network used the perfect shuffle interconnection, which makes the design compatible with our optoelectronic hardware designs.

This part of the work also included the gate-level design of the switching elements. We have done this using synthesis tools and figure 10 shows that the switching elements for synchronization require about 100 logic gates. This is an important factor in the design, because our present optoelectronic design is aimed at fine-grain switching elements. Figure 11 shows the gate-level design of the most complex switching element in our design.

4. FAULT TOLERANT RANDOM LIKE INTERCONNECTION

4.1 Expander graph algorithms

In this part of the research effort, we introduced the concept of parallel algorithms based on a random graph called expander graph and described its applications in optimal sorting algorithm, fault tolerant communication networks, associative memory, etc. We also proposed two optical interconnect approaches to realize such a system, one uses fixed interconnects based on computer generated holograms, while the other uses programmable interconnect based on photorefractive crystals (publication 3). We performed work on processing element (PE) designs that can realize the AKS sorting network. The AKS sorting network is an optimal sorting algorithm that uses expander graphs for interconnection. It represents a new class of architectures where no VLSI designs exist for comparison.

We found the following design tradeoffs for optical interconnect complexity, electronic complexity, and performance. The fully unfolded scheme maps the d interconnection stages into d different holograms, while the fully folded scheme substitutes that with only one stage of programmable interconnects. The unfolded scheme can use pipelining to greatly improve the system throughput and reduce the memory requirement on each PE (since each PE only needs to store the bits that are passing through that pipeline stage rather than the entire data packet). Although the folded scheme cannot pipeline the operations, it comes at a substantially reduced hardware cost since it only requires two stages of PEs (figure 12a)

The number of interconnection holograms can be reduced by increasing the number of detectors per PE. While the additional detectors also require associated amplifiers and logic for multiplexing. they help to reduce the number of stages and hence the interconnection storage requirement of the volume material. Consequently, optical throughput can be improved even when fewer reconfiguration is now required because between optical reconfiguration much sorting can be done electronically (figure 12b).

The main hurdle in implementing any expander graph based parallel algorithm is the number of stages of the interconnection. Recent progress has offered a compromise where only a fraction of the graph is generated randomly, and then extended in a deterministic manner.. This approach only involves two random interconnections from which further random permutations are generated deterministically. This reduction in the number of stages can be further improved by combining the two logical halves into the same PE plane, thus using half as many PEs compared to a fully folded approach. This dramatic reduction in hardware cost does not compromise the expansion of the graphs, which is the most crucial property of the network, although it does trade off throughput (figure 12c).

4.2 Twin butterfly network

Although the 2-permutation implementations would make the AKS network quite practical, the algorithm outperforms other sorting algorithms only for very large problem sizes. Furthermore it does not support pipelining and consequently have low throughput when used as an routing network. The twin butterfly network, on the other hand, can perform O(logN) routing in only logN stages of random permutation while still supporting pipelining. The twin butterfly is a multistage interconnection network that is based on the superposition of a normal butterfly on a permuted butterfly. Note that the butterfly is isomorphic to the perfect shuffle graph described earlier. The resulting network offers fault tolerance and low contention. In fact, the twin butterfly is a graph with weak expansion. In a typical interconnection network, the regularity of the network forces messages to share routes, resulting in higher delay. Furthermore, these networks do not have the ability to withstand failures. When more stages are added, they offer only limited fault tolerance. The random interconnect in twin butterfly distributes messages more evenly to reduce the network delay, and provides the network with alternate routes to go around faulty nodes (figure 13). Consequently the twin butterfly outperforms dilated butterfly, which is a modified butterfly network with comparable hardware cost (figure 14). We developed a set of software and use them in conjunction with a commercial package to simulate the performance of twin butterfly networks and compare them against others under different traffic load (figure 15). We then concentrated our efforts on the optoelectronic design and implementation of twin butterfly.

4.3 Design for testability

The main contributions of twin butterfly network are fault tolerance and low contention. Figure 14 graphically illustrates the ability to route a packet around faulty switching elements. In addition to performing the routing operation, the switch is designed to be tested both before and after the system is packaged and put into use. The block diagram of the switching element is shown in figure 16. Based on the tradeoff studies, we have reduced the number of control detectors in order to ease packaging. To tolerate faults during switch fabrication, the switches can be tested individually before fabricating the interconnection holograms. This requires modifying the switch design to support efficient testing. The main purpose of the modifications are to facilitate controllability and observability. The ability to control the internal states of a component through primary input, and the ability to observe the state change through primary output, together accomplish efficient testing. Note that this only helps tolerate faults detected before packaging. After the system is packaged, direct access to the modulator outputs of the internal switches is no longer available. In this case the PEs connected to the network can initiate built-in self-testing to mask the fault if possible, or perform automatic reconfiguration to route around faulty switches. Thus, additional logic is needed to perform system testing and reconfiguration after packaging.

In fabrication testing, a multiplexor is added to the input data path so that a signal arriving at a detector is routed to the modulator right away. By driving the four detectors with test patterns and stepping through the four detectors, we could detect faults in the detectors or the modulator by observing the modulator outputs. Since the input can be broadcast to all switching element on a chip simultaneously and observed in parallel on a CCD camera, we could test all switching element optically in parallel. Such optical testing significantly cuts down the testing time as well as the number of probes required to download the test patterns. With the well known stuck fault model, we could detect if any of the devices is stuck at logic 1, stuck at logic 0, or shorted to a neighboring device (bridging fault).

After a system is packaged, it becomes much harder to control the detector inputs directly. To reduce testing overhead, we designed testing logic into each switching element (This addition accounts for less than 1% of the routing logic). In system testing mode, each switching element will send test pattern to other switching elements through the two modulators. At the same time, it will compare the detector inputs (test pattern from other switching elements) with the test pattern

that it is putting out. A simple 1-bit comparator (2-input XOR gate) would be sufficient to detect a fault link and mark the corresponding flag to shut down this link.

System testing is carried out in two phases: forward, and then backward. Forward testing checks the data modulator and detectors while backward testing checks the acknowledgment modulators and detectors. The results are saved on registers on the switching elements and preserved through system resets.

The most critical aspect of twin butterfly's operation is the fault propagation and network reconfiguration. This is the time when a faulty switching element at stage *i* announces to the switching elements in stage *i-1* about its problems so that future packets will be routed to another switching element in stage *i*. If both destinations in stage *i* are found to be faulty, the switching element in stage *i-1* will propagate its problem to stage *i-2*. The reconfiguration takes logN steps since it takes that many cycles for the fault to propagate from last stage to the first stage. It turns out that our backward system testing procedure accomplishes the desired fault propagation. The network reconfiguration simply reduces to running system testing operation logN time. The hardware modifications to accomplish system testing and reconfiguration are shown in figure 17.

The reliability of the switching element has been analyzed in a combinatorial reliability model, assuming the well known exponential failure law. The model allows us to predict the availability of a switching element given failure rates for the components. The model could also be used in a top-down fashion to specify the device quality given a particular application that requires high availability.

4.4 Switching element control

We have designed the switching element to perform packet routing while supporting fabrication testing, system testing, and reconfiguration to mask out faulty devices. These different operation modes are controlled by two control signals. In the earlier design, we had the control signals passed from stage to stage, requiring two modulator-detector pairs on each switching element for this purpose. Since every switching element receives the same control signal, we could indeed broadcast the signal directly to all switching elements. This control scheme removes two modulators from each switching element and afford more area for silicon logic while improving yield. However, the two control detector scheme poses significant challenging in terms of packaging.

Packaging is significantly simplified by using only one single detector for sending the control signals. This is accomplished by using an asynchronous communication protocol for control bits, which sets the system modes between reset, system test, fabrication test, and normal routing (figure 18). The revised switching element layout is depicted in (figure 19). In previous layout, each modulator output is divided spatially among the four destinations. The quarter-size holographic optical element (HOE) for each link was a potential limit for scaling up to large networks. The revised arrangement gives the same HOE area for each link, significantly improving system scalability. The unused area could potentially be coated with reflective aluminum to use as bounce pads for long distance communications.

4.5 System simulation

We have built a full scale model of the POEM prototype in VHDL and developed a set of functional testing algorithms to demonstrate fault-tolerant operation on POEM (figures 20-21). Both fabrication and system testing operations have been modeled in VHDL. The first-in-first-out circular buffer has also been designed and simulated using B^2Logic software. Future work will involve layout, verification, and fabrication of the design for the entire switching element. This

work is detailed in publication 8. Space variant computer generated holographic optical elements will be used to implement the random interconnect and to map around defective switches found during fabrication testing.

5. CONCLUSIONS

During this project we designed an optoelectronic interconnection network based on the well known perfect shuffle network topology. Our work included the detailed design of the optoelectronic system, the layout of the optoelectronic chip, and the gate level design and simulation of the optoelectronic processing elements. To justify the implementation of the system, we have carried out a detailed comparison between our design and existing VLSI implementations and shown that optoelectronics outperforms VLSI for large network sizes. In addition, we have performed architectural and technological tradeoff studies to examine how various architecture and technology parameter variations affect the cost and performance of our design. Finally, we modified our basic system design to allow the ratio of optics and electronics in the system to vary without changing the system functionality. This allowed us to optimize the 2-D shuffle based optoelectronic MIN. The criteria for system comparison, tradeoff and grain size studies included system cost and performance functions such as system volume, system power consumption, on-chip power dissipation, and system bandwidth. The results indicated the optimized MIN would use 16x16 or 64x64 switches, corresponding to 250-400 transistors per optical I/O. We went beyond conventional perfect shuffle design, to implement a limited set of synchronization-type processing in the interconnection network. We have develop detailed designs of interconnection networks based on the shuffle interconnection topology that will support a complete set of communication and synchronization services in the network hardware.

We have also designed the PEs and the optical interconnection network for optoelectronic implementations based on random-like interconnection networks. We have studied engineering tradeoffs involve the balance between the grain size, the optical interconnect complexity, and the running time. It was found that networks based on random interconnects offer inherent tolerance to hardware faults in the processing elements. We have investigated the inherent fault tolerance of the network, and have also demonstrated the feasibility of parallel testing and run-time fault tolerance on a fully functional computer model of a POEM system. Work is in progress to develop algorithms for physical layout of the optoelectronic switches and reduce the complexity of the holographic optical interconnects. We are currently combining these results into an optoelectronic design and implementation of a fault-tolerant low-contention twin butterfly interconnection network.

6. LIST OF PUBLICATIONS

Journal papers

1. "Performance comparison between optoelectronic and VLSI multistage interconnection networks," F. E. Kiamilev, P. J. Marchand, A. V. Krishnamoorthy, S. C. Esener, and S. H. Lee, *IEEE J. Lightwave Technology* Vol. 9, No. 12, pp. 1674-1692 (1991).

2. "Grain-size considerations for optoelectronic multistage interconnection networks," A. V. Krishnamoorthy, P. J. Marchand, F. E. Kiamilev, and S. C. Esener, *Applied Optics* Vol. 31, No. 26, pp. 5480-5507, (1992).

3. "Parallel algorithms based on expander graphs for optical computing," R. Paturi, D.-T. Lu, J. E. Ford, S. C. Esener, and S. H. Lee, *Applied Optics* Vol. 30, No. 8, pp. 917-927 (1991).

4. "Design tradeoffs in optoelectronic parallel processing systems using smart-SLMs," D.-T. Lu, V. H. Özguz, P. J. Marchand, A. V. Krishnamoorthy, F. E. Kiamilev, R. Paturi, S. H. Lee, and S. C. Esener, *Journal of Optics and Quantum Electronics* Vol. 24, pp. 379-403, (1992).

Conference papers

5. "Design of interconnection networks for programmable optoelectronic multiprocessors," F. Kiamilev, P. Marchand A. Krishnamoorthy, K. Urquhart, R. Paturi, S. Esener, and S. H. Lee, *OSA Topical Meeting on Optical Computing*, Kobe, Japan, April (1990).

5. "Grain-size considerations for programmable optoelectronic multiprocessor technology," F. Kiamilev, A. Krishnamoorthy, and P. Marchand, K. Urquhart, and S. Esener *Technical Digest OSA Annual Meeting*, Boston, November (1990).

7. "Effects of optoelectronic device characteristics on the performance and design of POEM systems," S. Esener, F. Kiamilev, A. Krishnamoorthy, and P. Marchand, *Proc. SPIE Annual Meeting*, paper 1562-02, San Diego, July (1991).

8. "Architectural and technological tradeoffs for optoelectronic multistage interconnection networks", F. Kiamilev, A. Krishnamoorthy, P. Marchand, S. Esener, and S. H. Lee, San Jose, California (1991).

9. "An optoelectronic system for distributed computing", F. Kiamilev, A. Krishnamoorthy, and S. Esener, *Technical Digest OSA Annual Meeting*, San Jose, California (1991)..

10. "Smart pixels for 2-butterfly interconnection network", D.-T. Lu, R. Paturi, S. Fainman, and S. H. Lee, to be presented in OSA Annual Meeting, San Jose, California (1991).

11. "Fault-tolerant computing on POEM", D.-T. Lu, T. Y. Lin, F. Kiamilev, S. C. Esener, and S. H. Lee, OSA Topical Meeting on Optical Computing, Salt Lake City, Utah (1991).

12. "Optoelectronic Module Architecture for Smart Switching Networks", F. Kiamilev, A. Krishnamoorthy, and S. Esener, *Proc. SPIE Symposium on Optoelectronic Packaging and Interconnects*, paper 1849-19, Los Angeles, (January 1993).

7. TABLES AND FIGURES

Routing Performance of N=1024 2-Butterflies

Number of packet routing delays in the presence of increasing faults:

	normal	dilated	2-butterfly			
no. of faults	0	0	0	10	100	1000
1 random permutation	14	12	12	12	12	14
10 random permutations	25	19	18	18	22	35
1 transpose permutation	30	17	11	11	11	13
10 transpose permutations	269	157	19	18	22	36

Table 1: Routing performance of twin butterfly compared to regular butterfly and dilated butterfly. Notice that twin butterfly has comparable or better performance without faults.

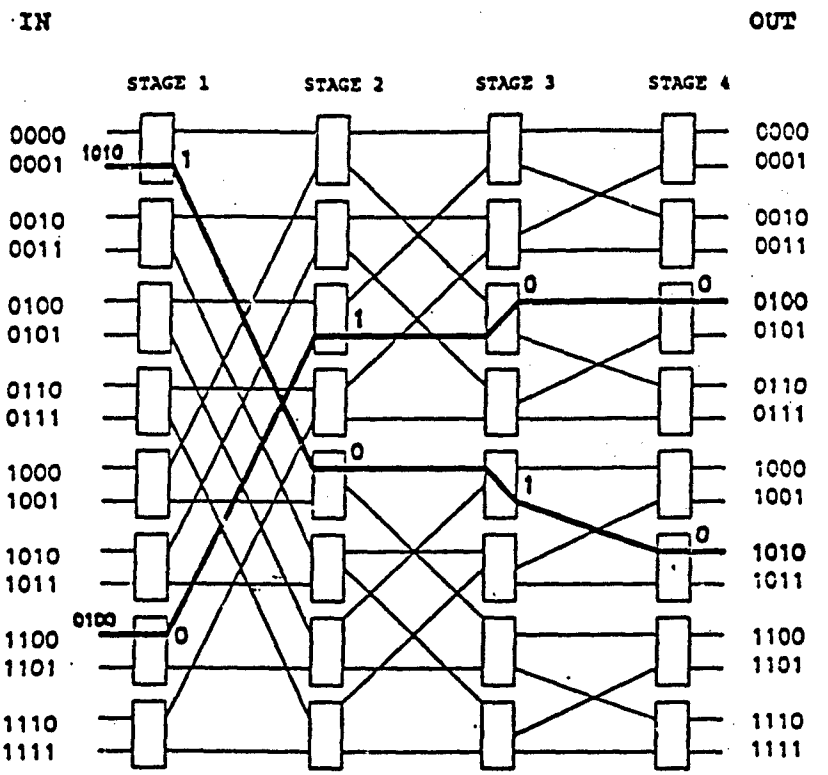


Figure 1. A Banyan interconnection network with 16 channels. The highlighted paths illustrate the destination based routing algorithm.

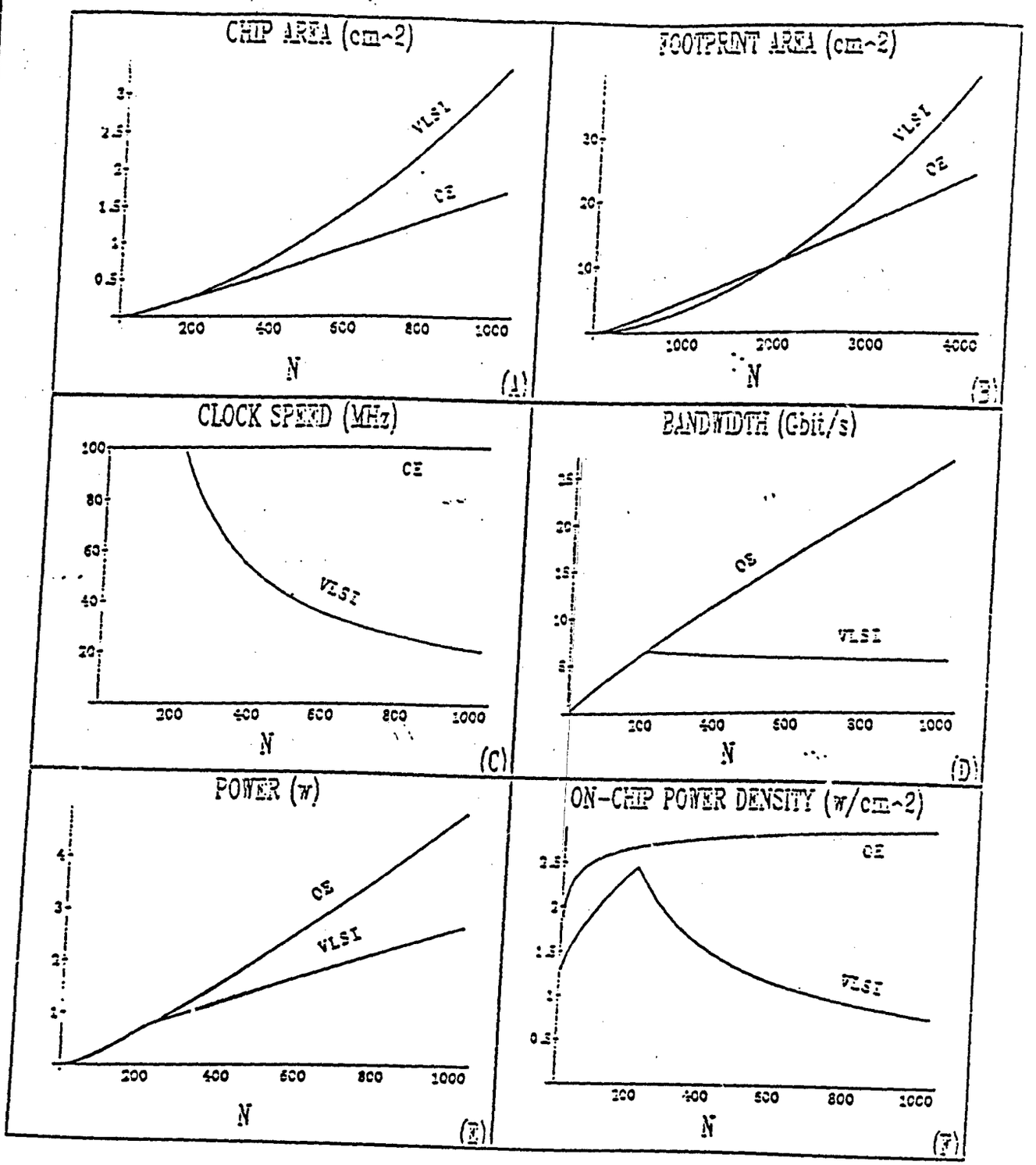


Figure 2 : Quantitative comparisons between optoelectronic and VLSI multistage interconnection networks.

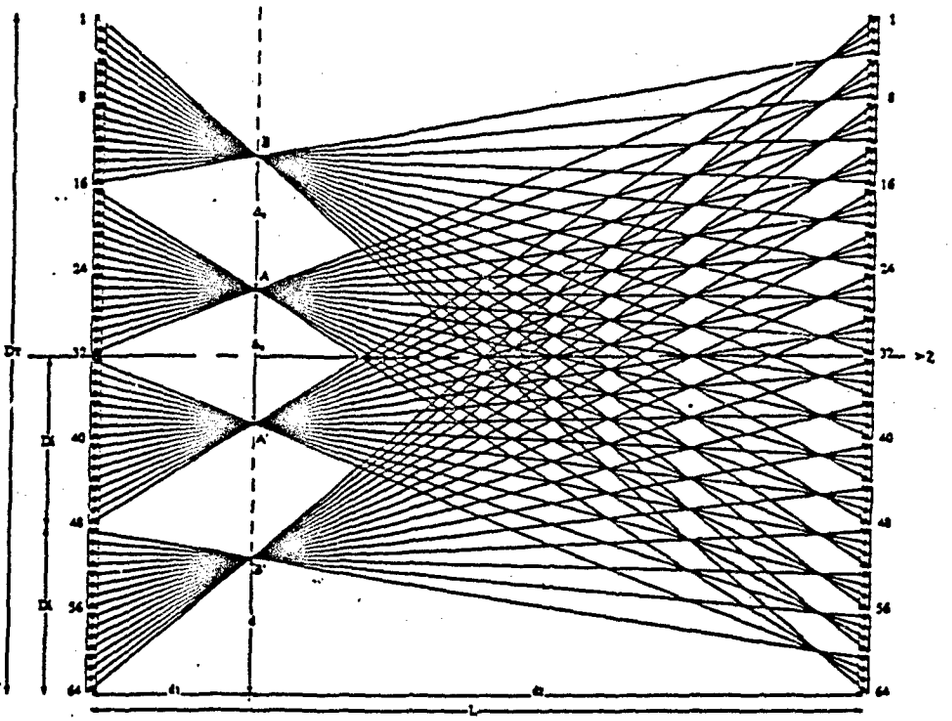


Figure 3. One dimensional view of one stage of an N=4096 channel, K=16 grain-size shuffle-exchange interconnection.

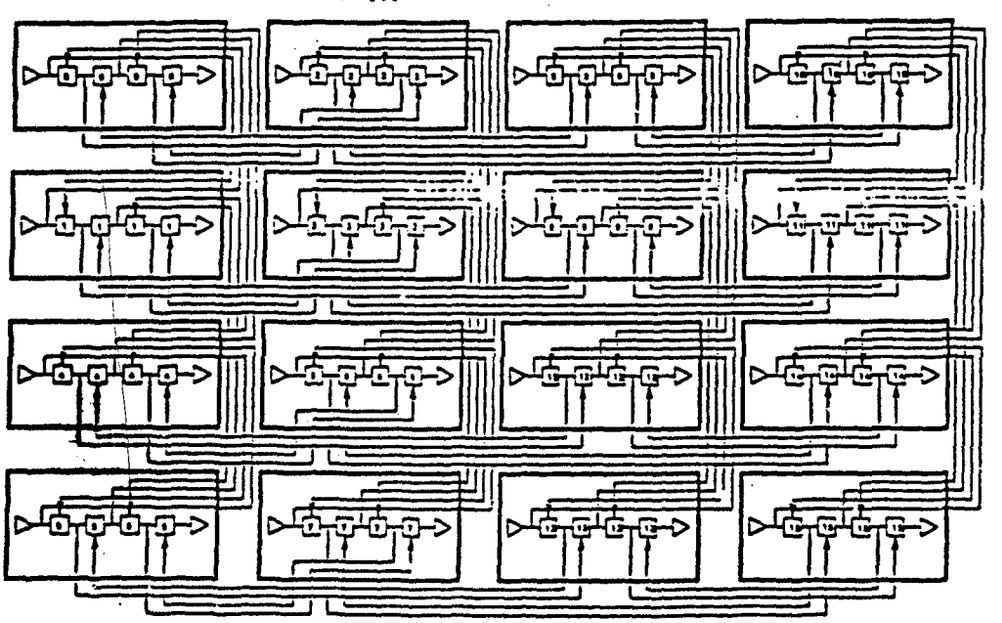


Figure 4. 2-D electronic layout for N=16 channel shuffle network. Modulators and detectors are evenly distributed in the grain.

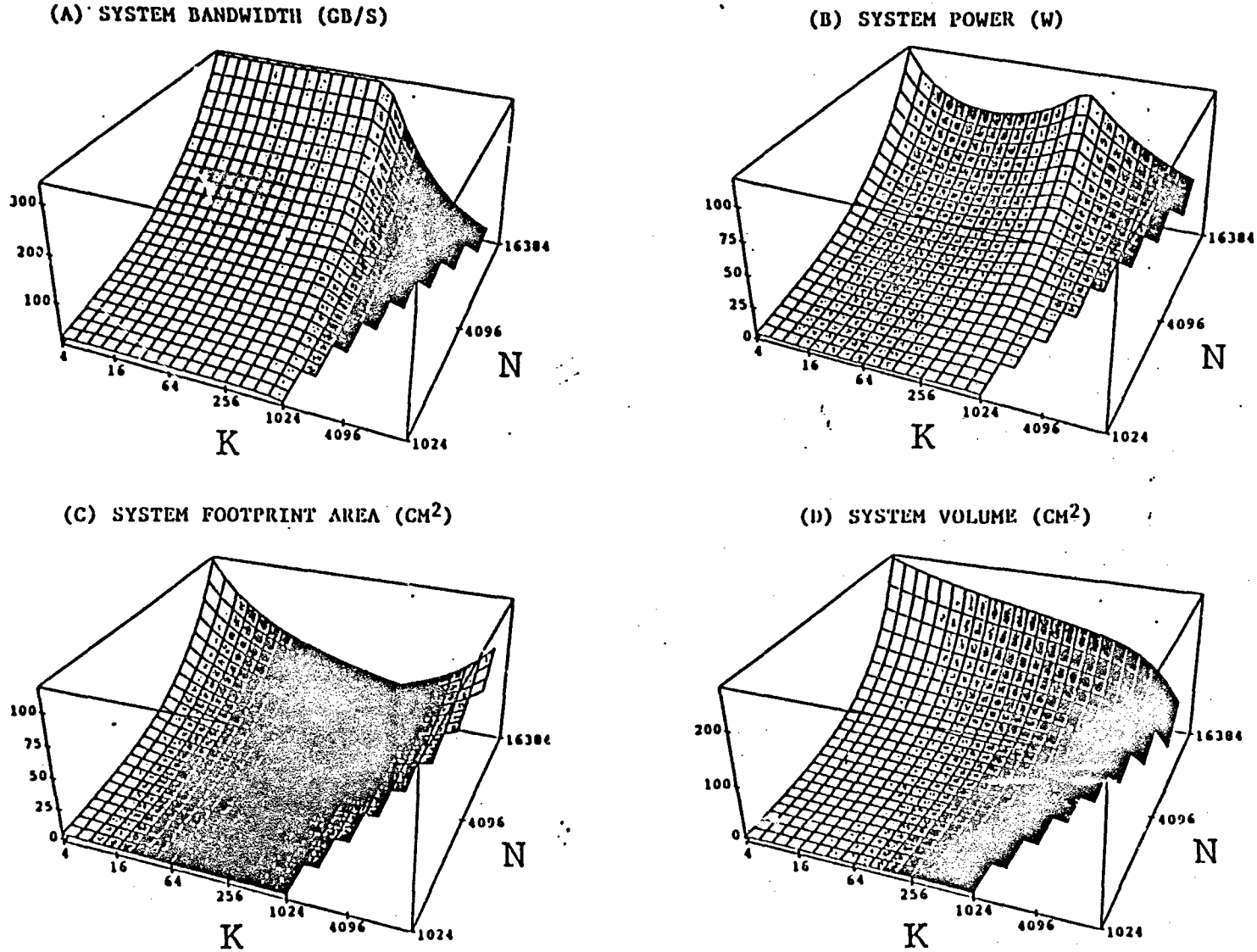


Figure 5. 3-D plots of the basic grain-size study results showing: (a) system bandwidth, (b) system power, (c) system area, (d) system volume. Note that the graphs are valid only at integer values of N and K, where K is a power of 4.

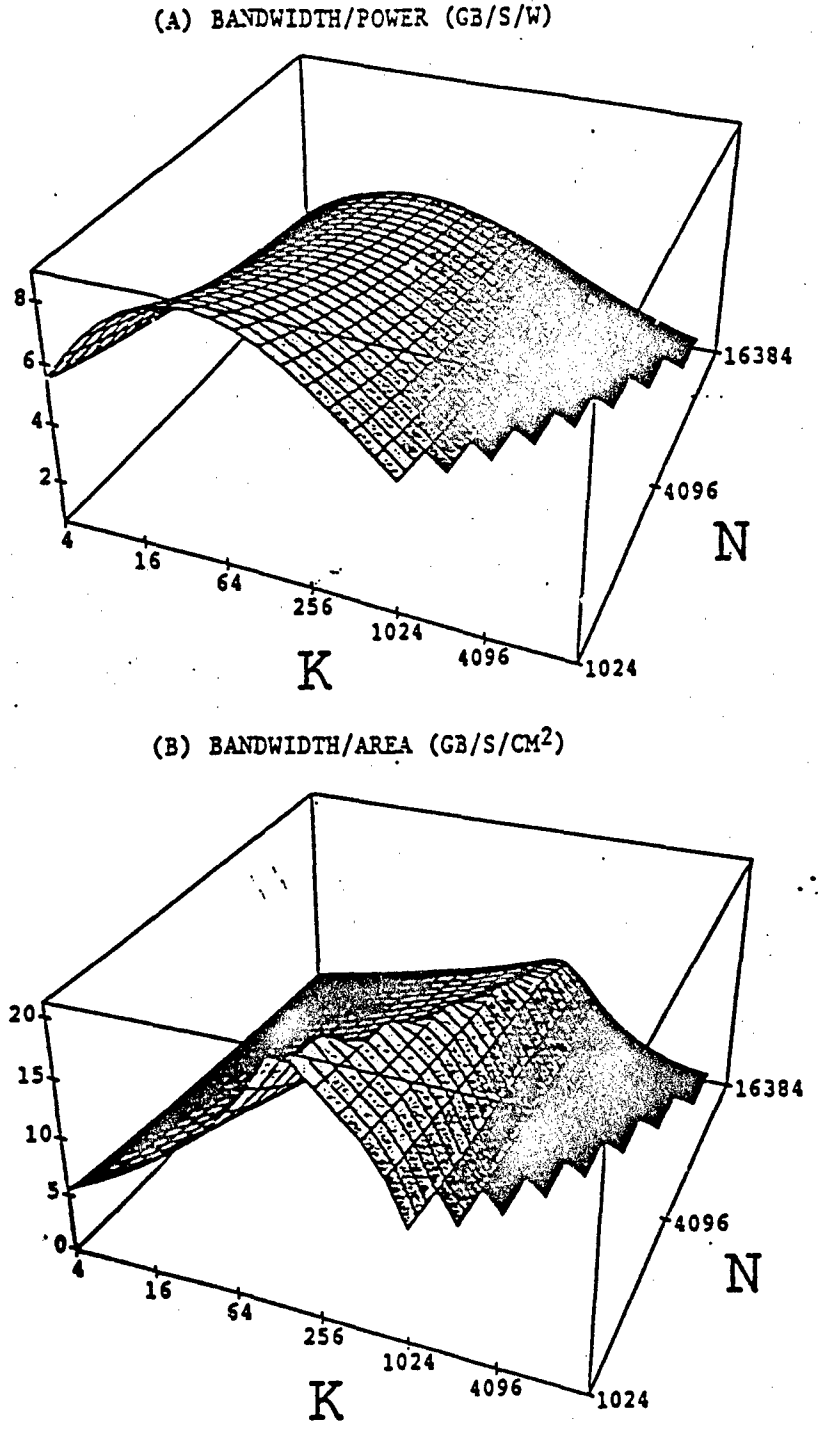


Figure 6. Performance/cost metrics for the 2-D shuffle-exchange network with variable grain-size as a function of system size (N) and grain-size (K): (a) bandwidth/power, (b) bandwidth/area

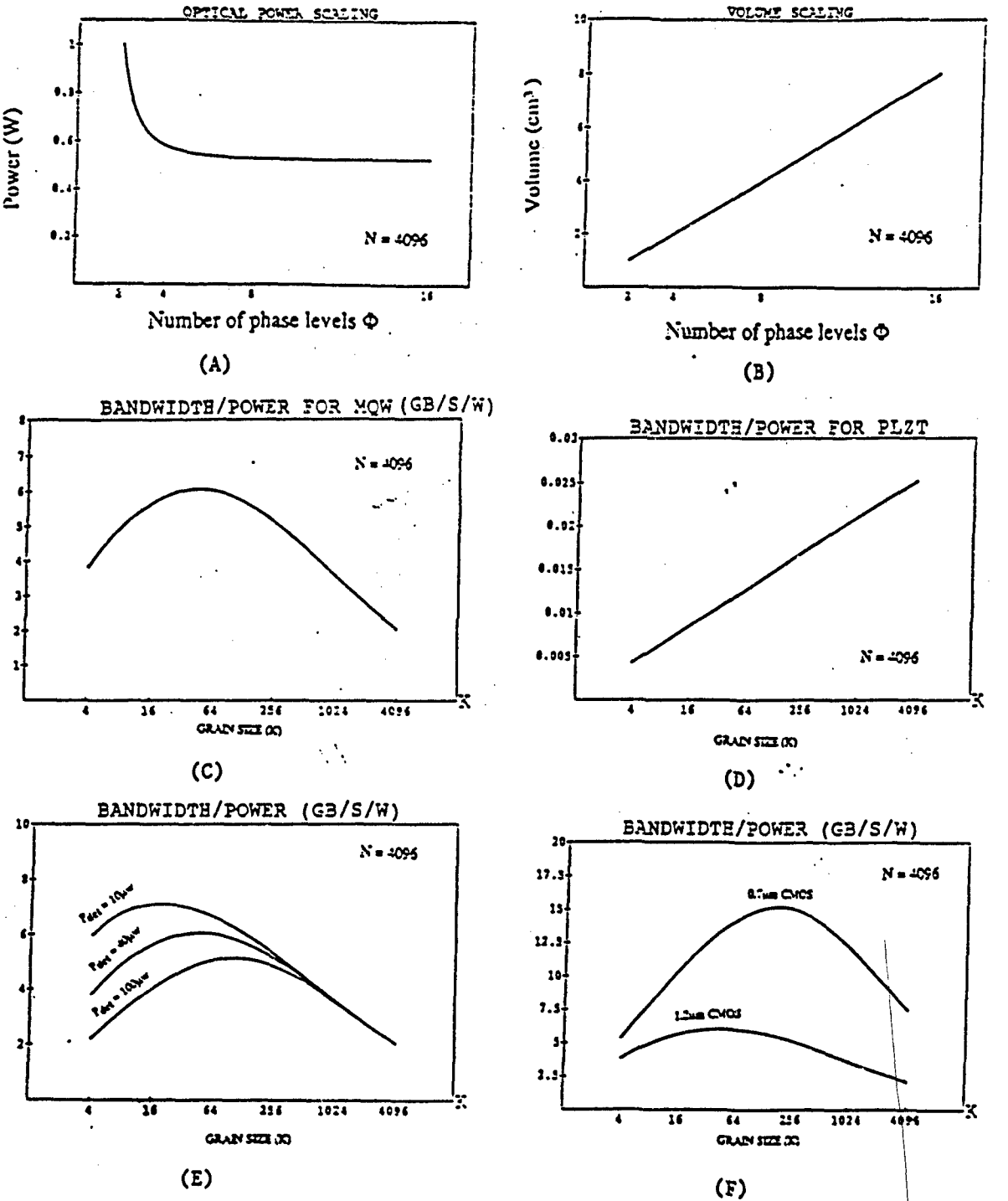


Figure 7. Effects of optoelectronic device characteristics on system performance: (a) optical power vs. # phase levels, (b) system volume vs. #phase levels, (c) bw/power for MQW, (d) bw/power for PLZT, (e) bw/power vs. min. detectable power, (f) bw/power for 0.7um vs. 1.2um CMOS VLSI technology.

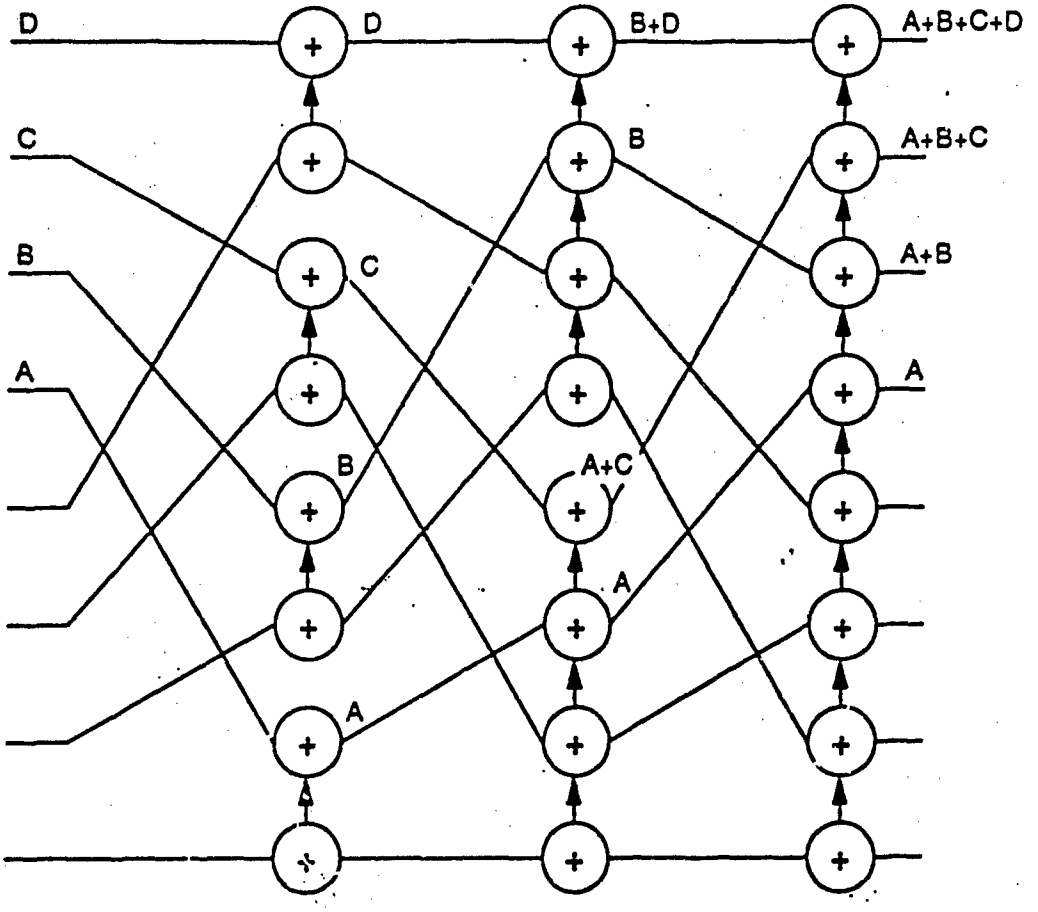


Figure 8: Topology of the distributed ADD network. This figure shows that the processing networks use the shuffle topology between stages and require additional near-neighbor interconnections within the stage. The local connections are used to determine whether the switching element should perform the ADD operation based on the contents of the incoming packet and the packet of its neighbor switching element.

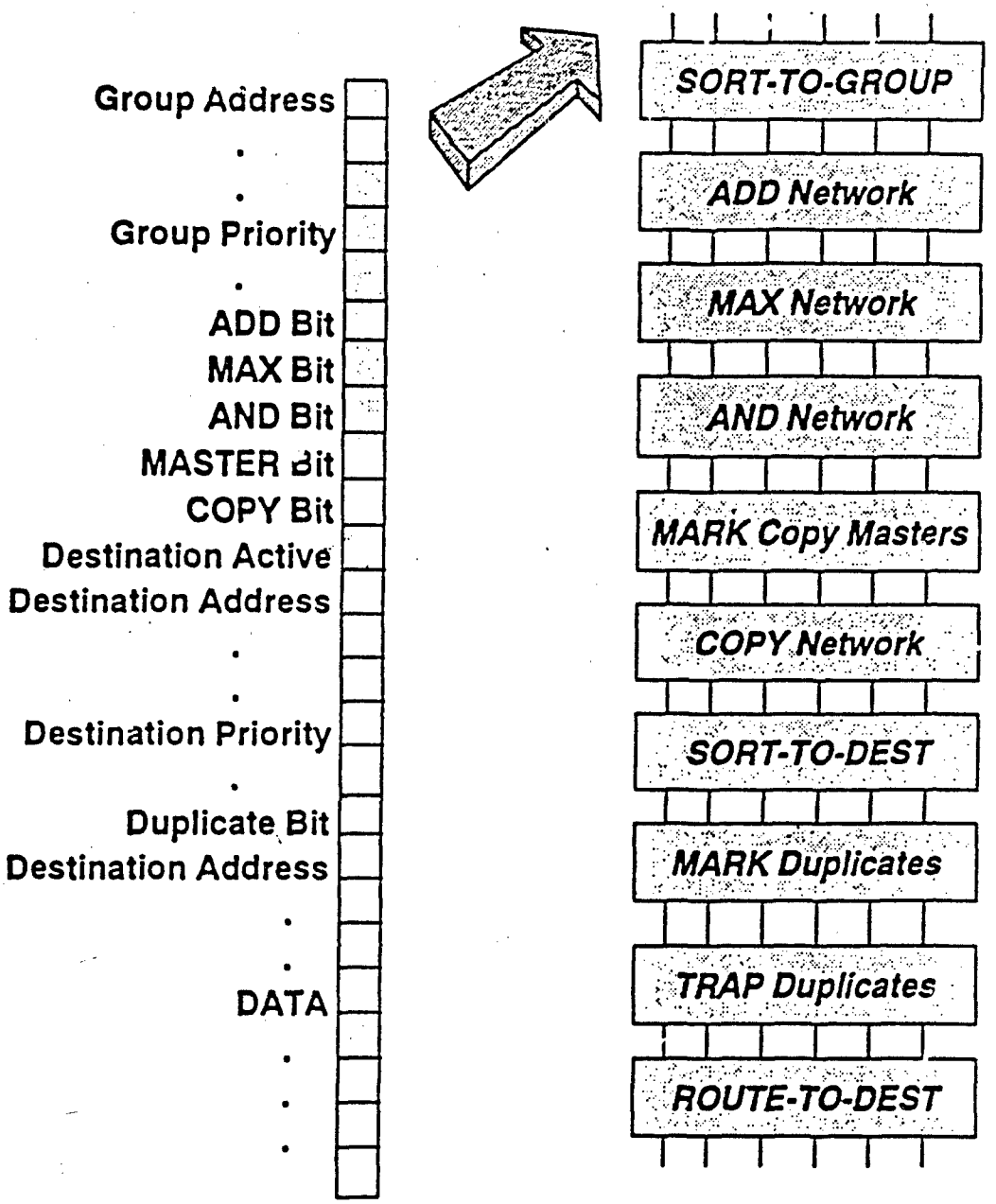


Figure 9: Smart network packet format. This figure shows the packet format and the smart network subnetworks.

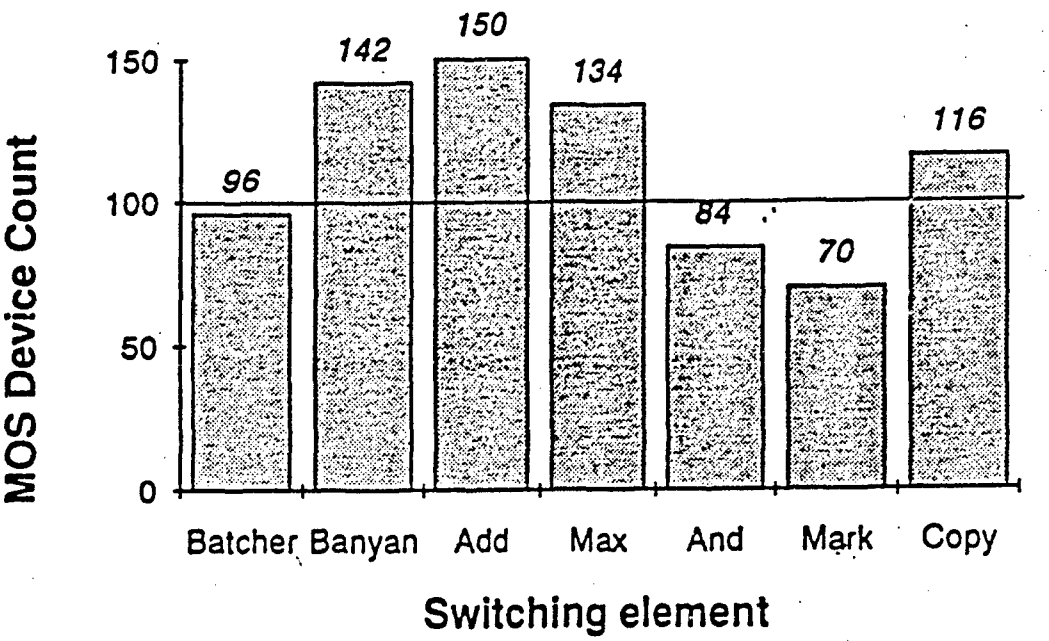


Figure 10: Histogram of gate count for smart min switches. This histogram shows that the switch gate count is near 100 MOS devices for all the switching elements in the smart min. This property enables efficient optoelectronic implementation.

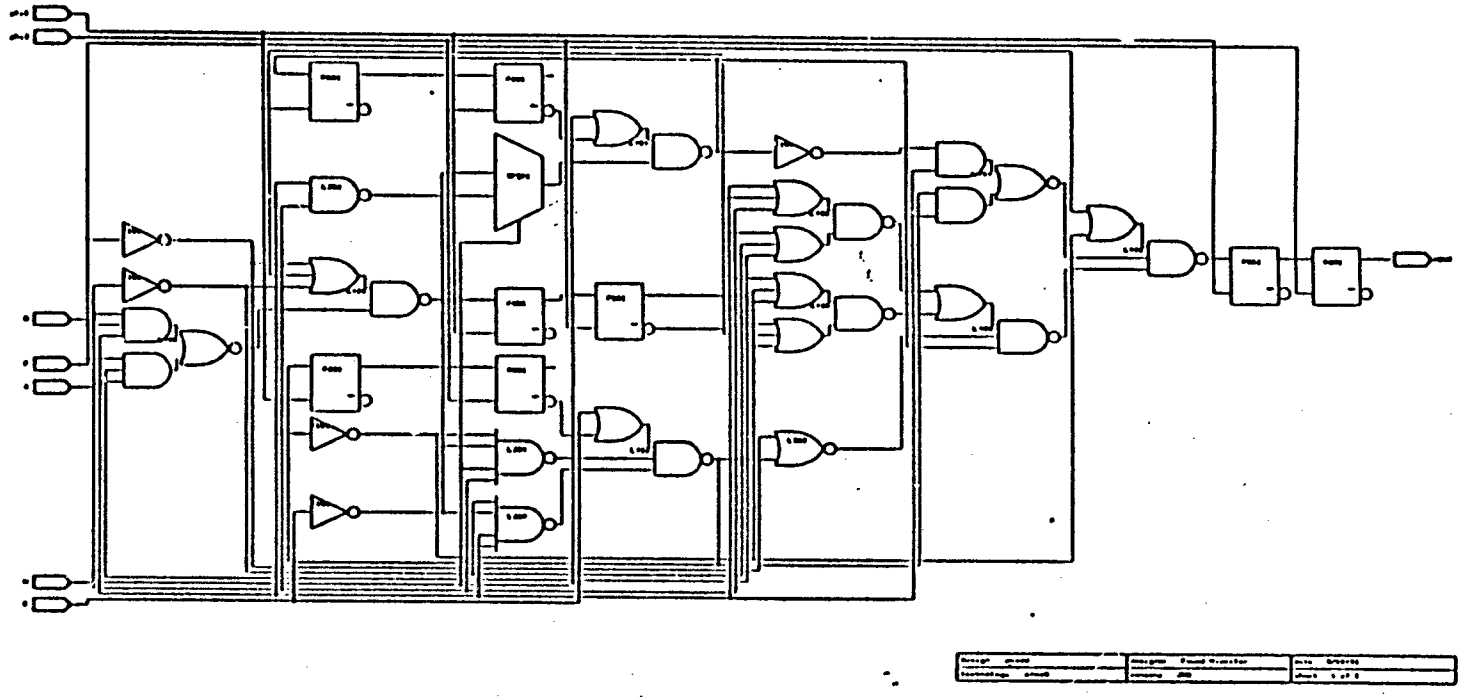


Figure 11:. Distributed ADD network switching element.

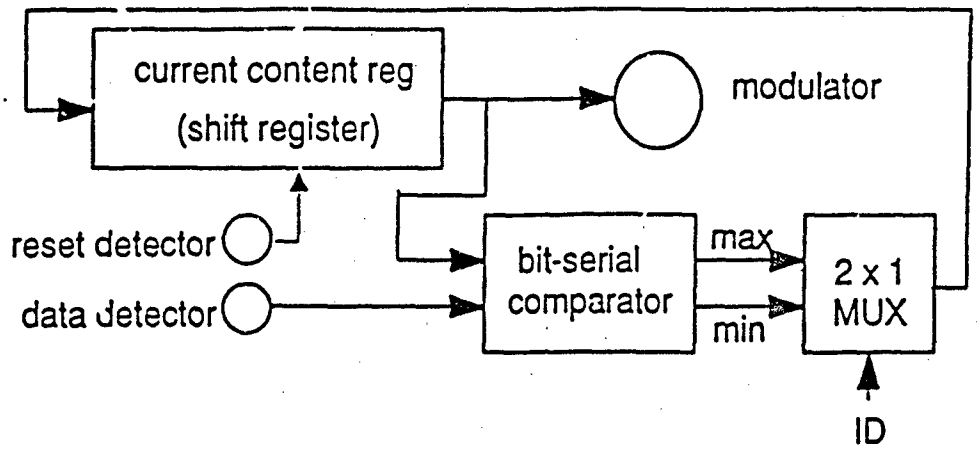


Fig.12a) Basic functional design of an expander graph PE for folded setup.

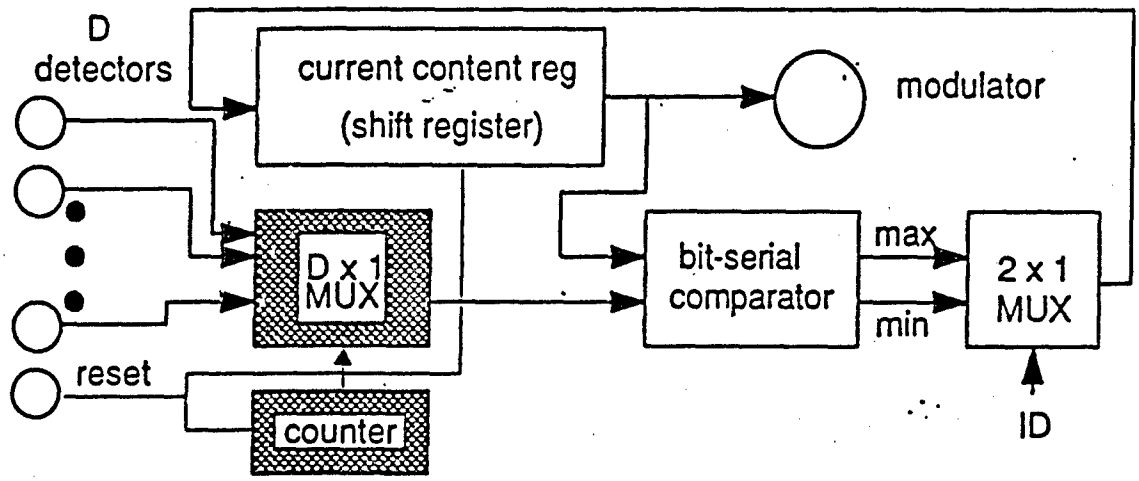


Fig.12b) Multiple detector functional design. The shaded components indicates components that scale with D. The unshaded ones remain the same regardless of D.

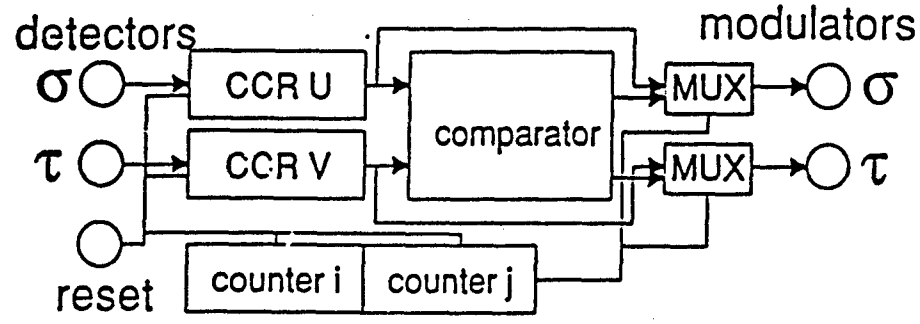


Fig.12c) PE design for 2-permutation expander. σ and θ denote the two random interconnects used. U and V denotes the two (logical) processing planes. This design only requires half as many PEs.

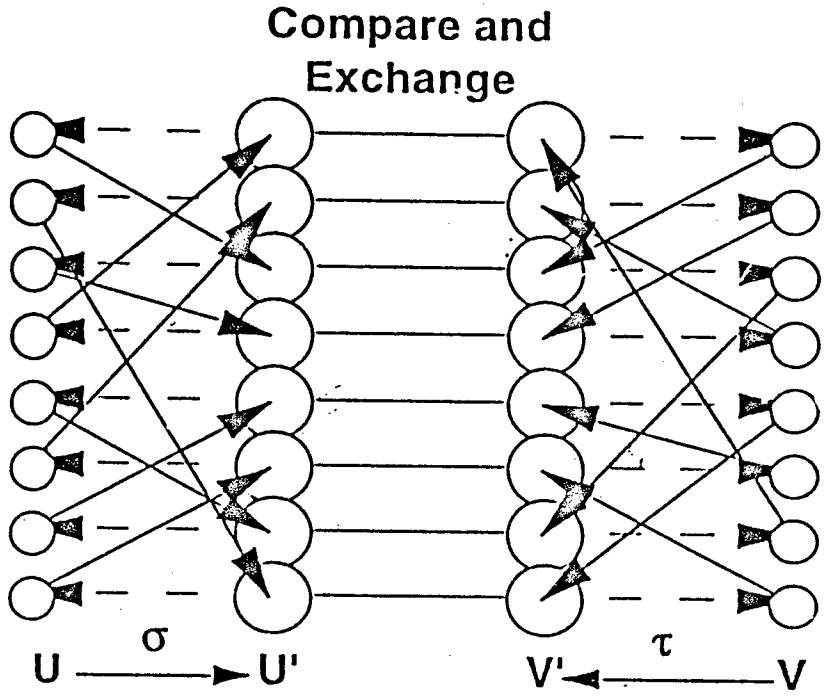
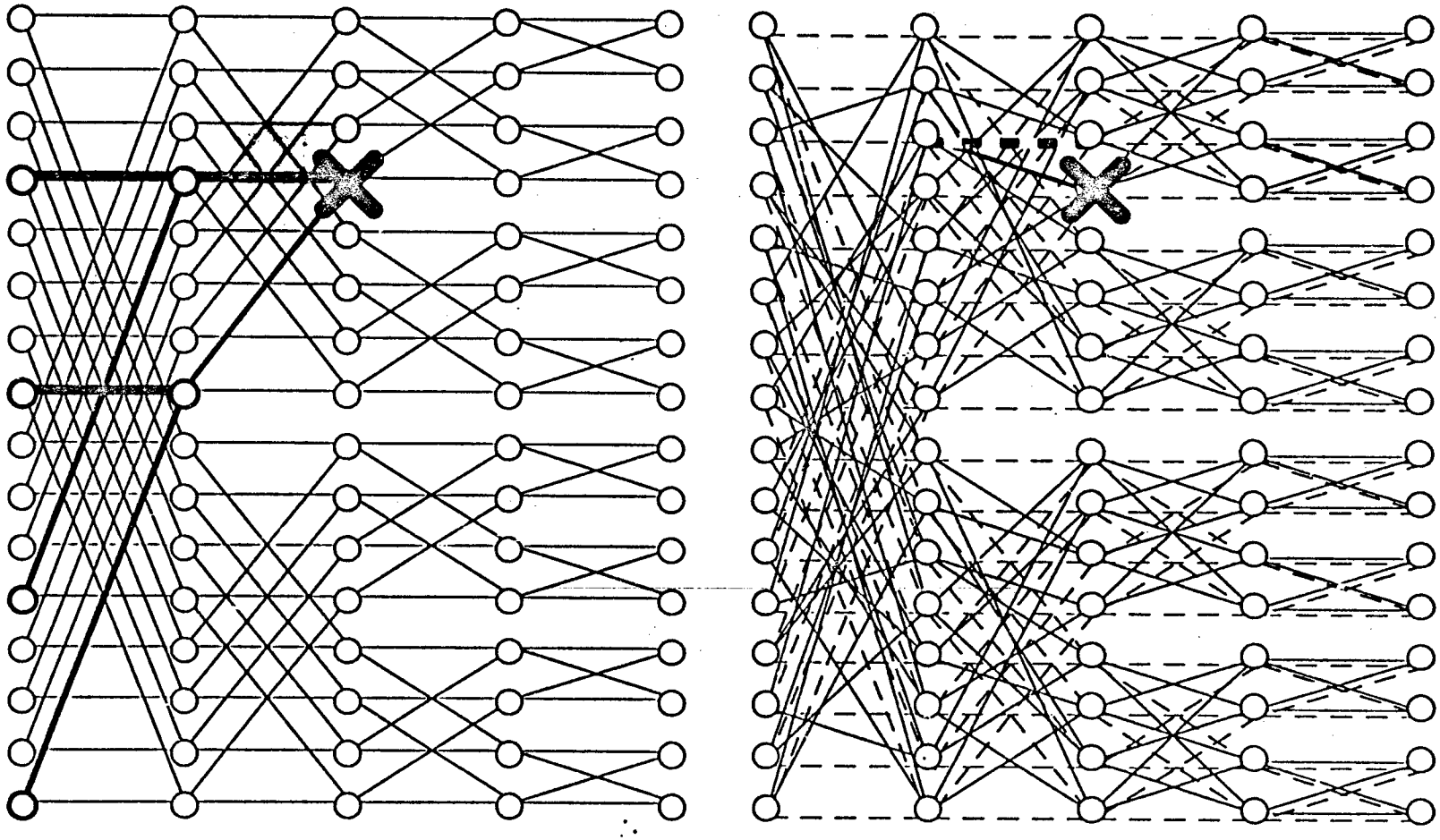


Figure13: 2-permutation expander. The random permutations, σ and τ, are used to generated the random permutations $\sigma\tau$, $\sigma^2\tau^2$, $\sigma^3\tau^3$, $\sigma^4\tau^4$, etc.



Figure|4: Twin butterfly fault proliferation. The dilated butterfly can tolerate single link failures since all links are duplicated. It cannot handle the more severe switch failure that propagates backward to block more inputs. In twin butterfly, messages can route around faulty switches to *mask* the fault.

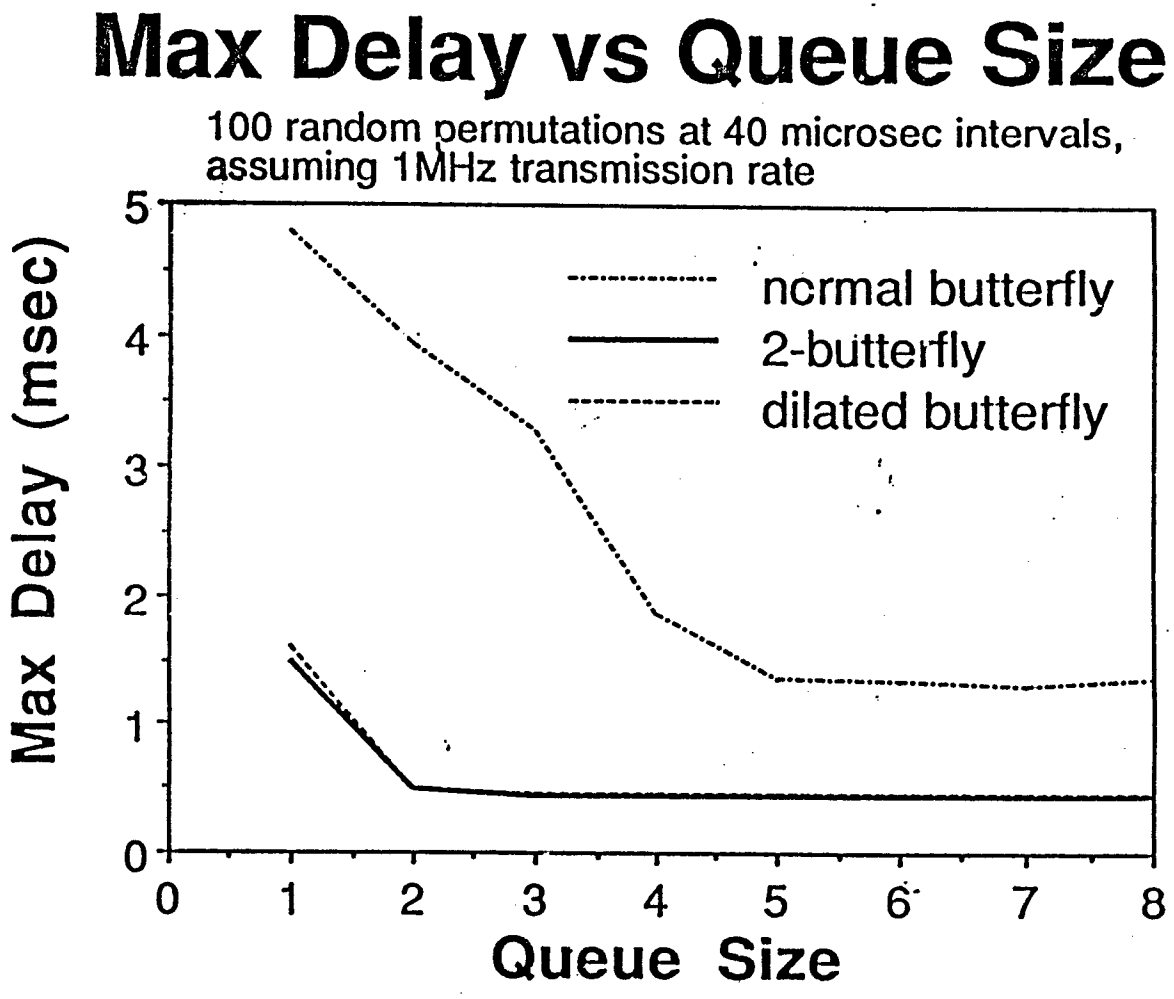


Figure15: The maximum network delay of twin butterfly compared with dilated and regular butterfly. The input is defined to be 100 consecutive random permutations. While twin and dilated butterfly have comparable performance for this input, the simulation has assumed that all switches are perfectly fabricated and correctly functioning all the time. Future work will simulate fault tolerance operation as well.

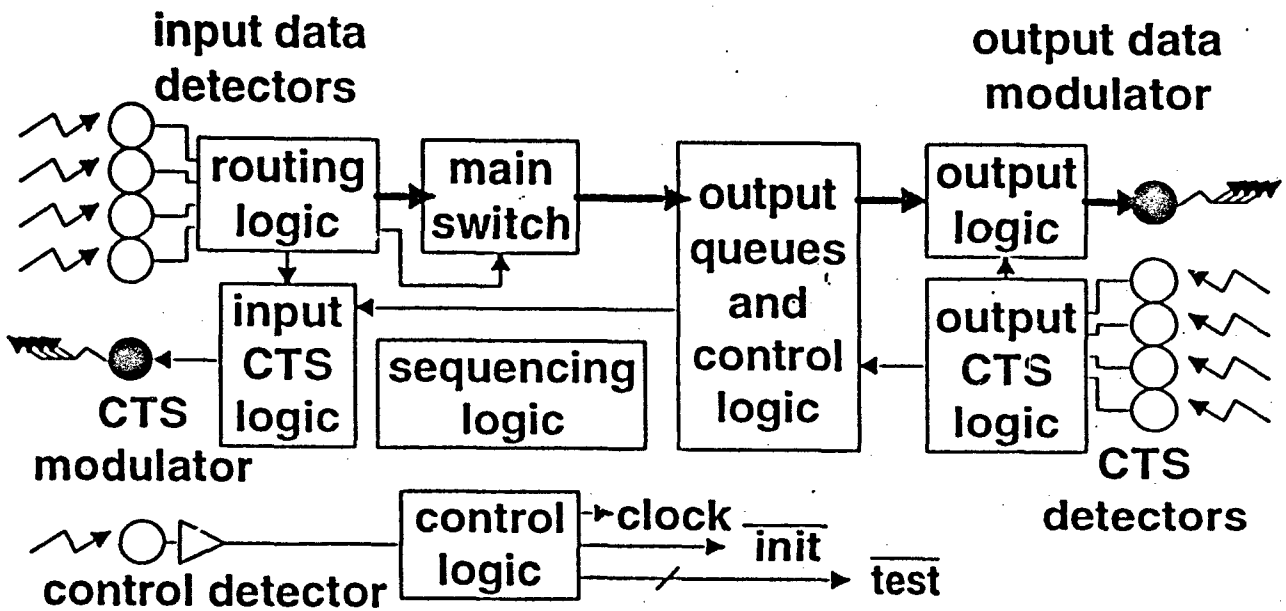


Figure 16: Switching Element Block Diagram

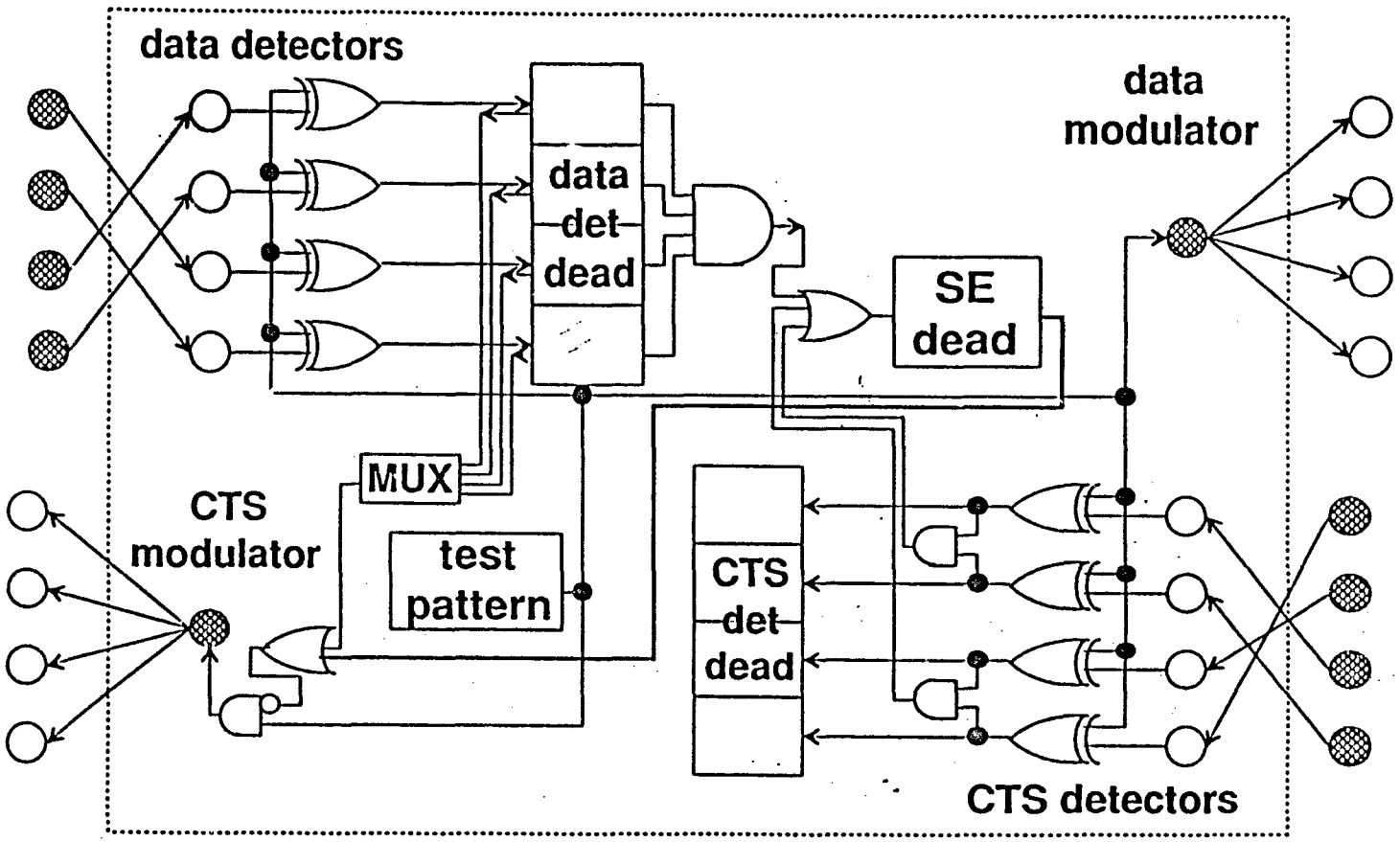


Figure 17: System testing and reconfiguration. Faults propagate from last stage to first stage. Faulty switching elements appear as CTS detectors stuck-at-0. Reconfiguration completed in LogN steps.

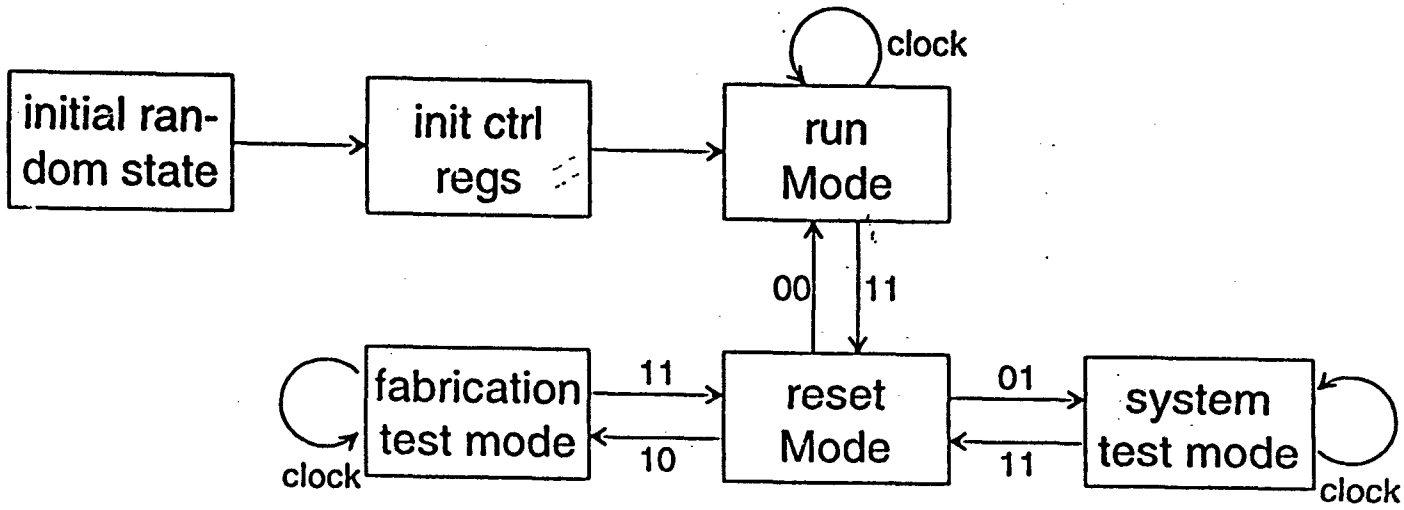


Figure 18: Control Logic. Control registers are initialized at power-up via electrical interconnect. During test modes, clocking increments the 2-bit counter to sequence through the four detectors.

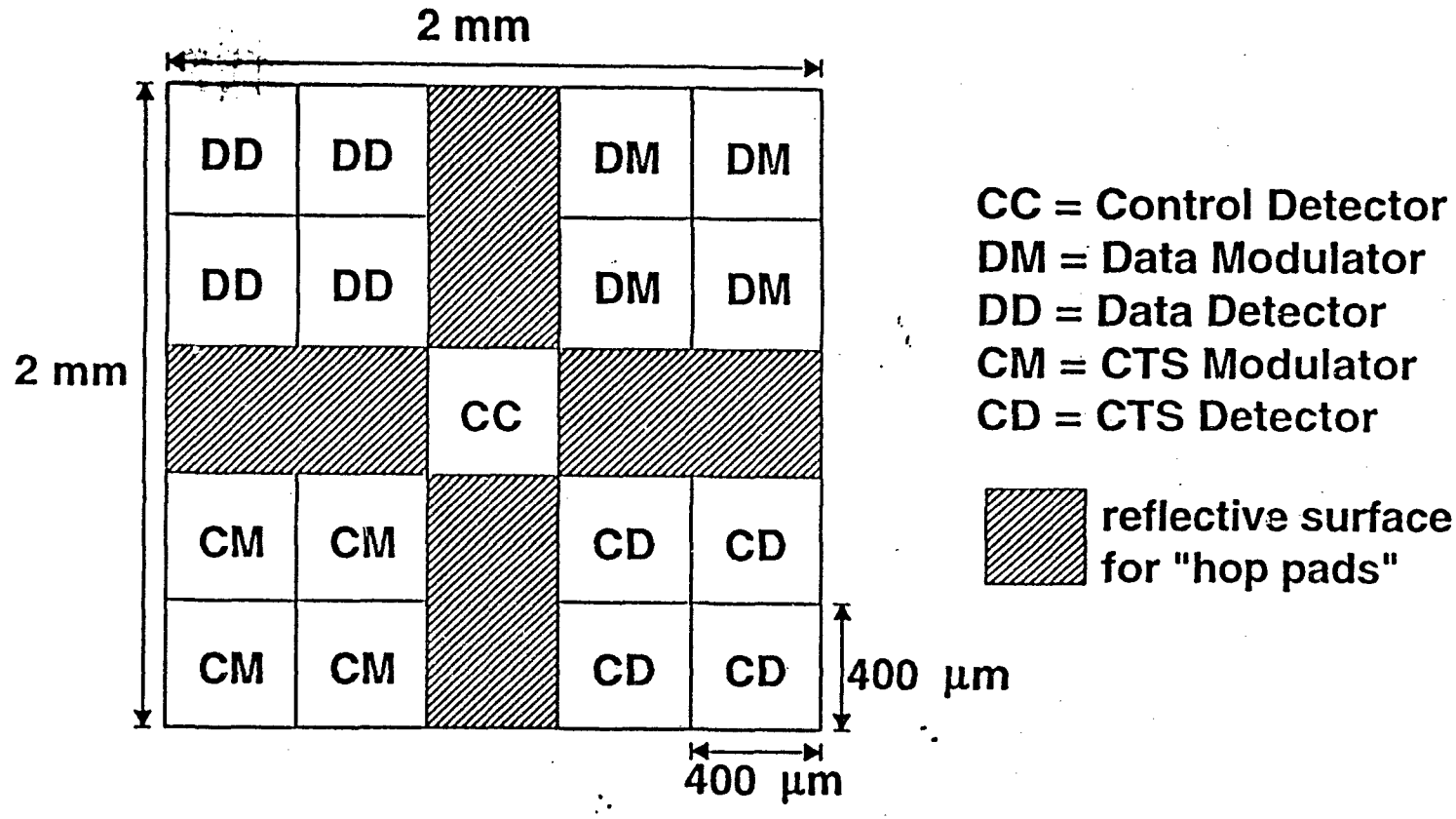
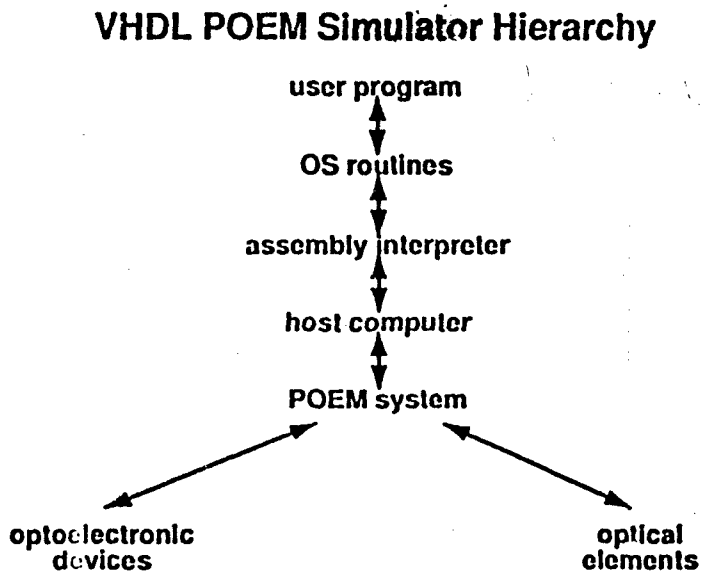


Figure 19: Switching element layout. Layout minimizes electronic wiring. Larger SLM area improves scalability. Only one control detector eases packaging.



⇐ user program written in POEM mnemonic assembly and operating system calls

⇐ OS routines to initialize system, clear ESLM, load ESLM, read detector array, initiate check point testing and reconfiguration, etc.

⇐ interpreter for the POEM assembly language

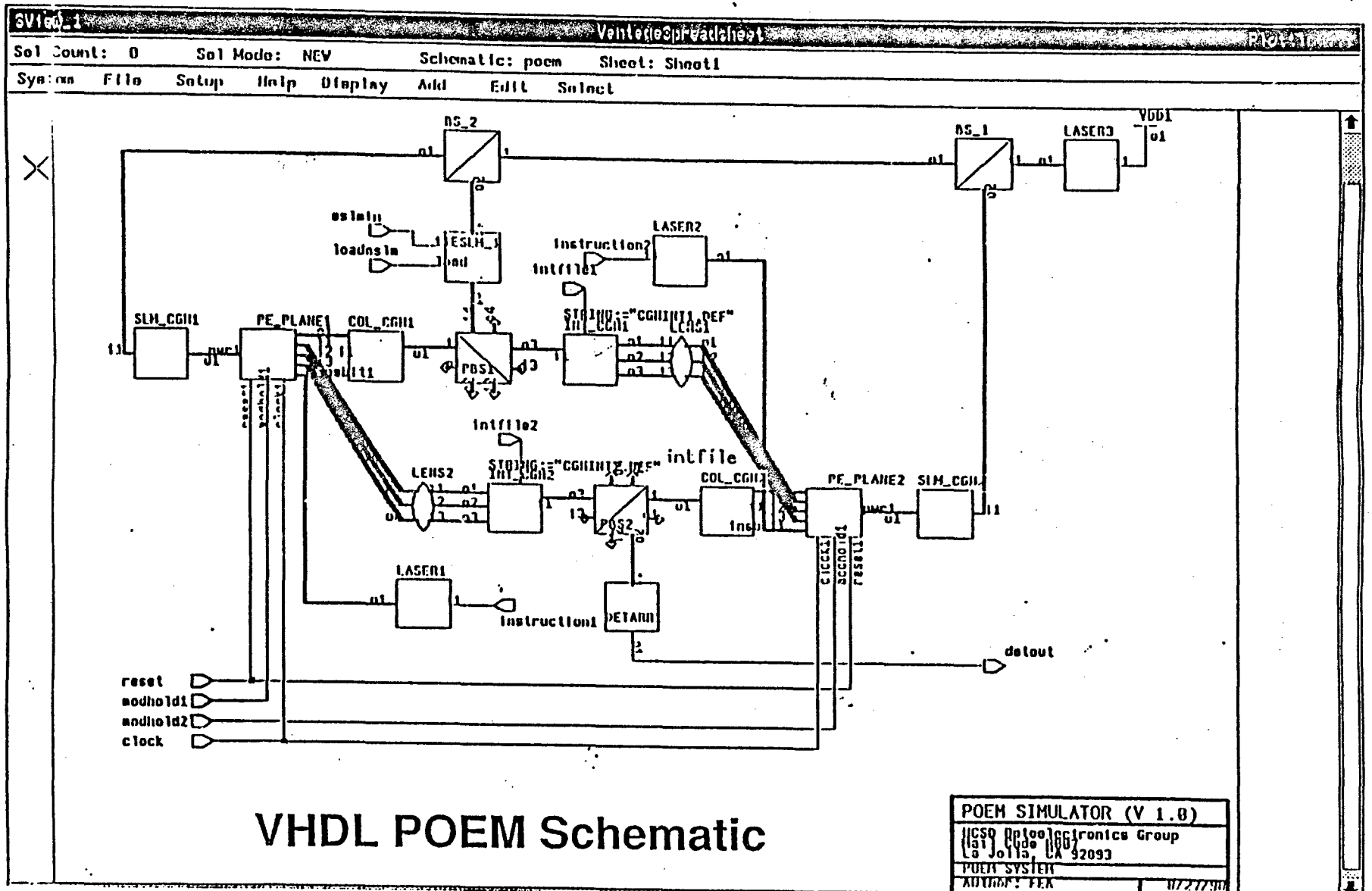
⇐ behavioral model of a microcomputer that interfaces between POEM hardware and software

⇐ collection of two processing planes and optical interconnection using CGH

⇐ behavioral models of lens, PBS, laser source, collimating CGH and interconnection CGH

⇐ behavioral models of detector, SLM, and the CMOS PE(1 modulator, 3 detectors, 1-bit ALU and 64 bit RAM)

Figure 20: VHDL POEM Simulator Hierarchy. Similar to a modern digital computer architecture, the POEM model is built on top of behavioral models of optoelectronic devices and optical elements. User programs only see the assembly language level and the OS service routines that can be added into the host computer to provide more functionality.



VHDL POEM Schematic

Figure 21: VHDL POEM Schematic. The simulation layout corresponds directly to the optical system layout on an optical table. The labeled ports represent the interface to the host computer.